

在线学习平台集群服务器架构设计与实现*

李宁¹ 左悦²

(1) 南宁学院信息工程学院, 南宁, 530200 (2) 南宁学院土木与建筑工程学院, 南宁, 530200

摘要 随着在线学习服务器访问量大大增加, 增大了服务器的压力, 对实时性、稳定性要求高的在线评分功能受到影响。经分析, 后台服务器存在单点故障风险, 主要有 Web 服务端的单点故障、数据库服务端的单点故障和文件存储服务端的单点故障、架构的耦合度过于密集等, 这些对数据安全、在线学习业务正常运行造成很大的威胁, 数据一旦丢失, 带来不可估计的损失。针对该问题, 利用现有的技术设计了一套新的平台集群架构方案, 使得服务器具有高性能、高可用、可伸缩及支持高并发访问的优势。架构不仅是有效地解决了目前在线学习平台服务端架构的数据安全及正常运行等问题, 也为中小型企业提供高可用网站架构部署的参考。

关键字 在线学习平台, 集群架构, 服务器, 数据库

Design and Implementation of Online Learning System Cluster Server Architecture

Li Ning

School of Information Engineering
Nanning University
Nanning 530200, China
bgy_2009@163.com

Zuo Yue

School of Information Engineering
Nanning University
Nanning 530200, China
gladyz5487@126.com

Abstract—With the great increase of online learning server access, the pressure of the server is increased, and the online scoring function with high real-time and stability requirements is affected. After analysis, the background server single point fault risk, mainly Web server single point fault, single point fault and database server file storage server single point fault, architecture coupling degree is too dense, etc., the data security, online learning business running is a great threat, data once lost, bring immeasurable loss. For this problem, a new set of platform cluster architecture scheme is designed using the existing technologies, making the server has high performance, high availability, scalability, and support of high concurrent access advantages. The architecture not only effectively solves the problems of data security and normal operation of the current online learning system server architecture, but also provides a reference for small and medium-sized enterprises to use the deployment of high-available website architecture.

Key words—Online learning system, Cluster architecture, Server, Database

1 服务器主要存在的问题分析

分析和研究现有的在线学习平台集群服务器的应用状况, 发现服务器主要存在以下主要问题:

(1) 服务器运行饱和

服务器架构可扩展性差, 随着在线用户的请求量不断增长, 服务器端的部分节点将会达到性能饱和状态, 从而将会造成整体架构的稳定性松动, 如果分流

不及时, 内存、CPU、I/O 等资源使用率较高时, 整个服务器存在宕机的风险。

(2) 数据安全性

当用户数据写入时, 将会保存在后台的文件存储服务或数据库服务器上, 服务器在高负载的情况下, 很容易造成用户数据的丢失, 也有可能出现服务器系统崩溃无法恢复的风险; 用户数据是无价的, 服务器的设计应保证数据的安全性。

(3) 用户体验差

服务端的压力将会影响到前端用户的体验, 数据服务器的处理事务速度值降低时, 将会影响到数据查

*基金资助: 本文得到广西高校中青年教师科研基础能力提升项目(2021KY1805)资助。

**通讯作者: 左悦, 副教授, gladyz5487@126.com

询的性能，数据库应对不了大量的查询请求，将会有大量的请求在排队中，返回给前端用户的时间也随之变长，影响客户的体验。

（4）评分业务的实时性差

在校园网路中，学生提交作业后，由于服务器端的超负荷运行，CPU、内存等资源得短缺或者分配不合理等因素造成结果返回实时性较差^[1]。

上述服务器架构存在普遍的单点故障、服务软件之间的耦合度高等问题，十分容易造成服务器因压力过大而造成宕机的风险，因此需要对现有的服务端架构进行改造和优化。

针对以上问题，本文将设计一套新的平台集群架构方案，以发挥服务器具有的高性能、高可用、可伸缩及支持高并发访问等技术优势。

2 在线学习平台集群服务器架构的需求

在线学习平台集群服务器架构的需求主要有以下方面：

（1）学习平台 7*24 小时不宕机

服务器 7*24 小时不宕机的范围相对有点宽些，它不仅仅是指在物理服务器的故障和业务故障时可以快速恢复，还应有在网站流量激增时，网站依然可以高效率地处理用户发来的请求业务。

（2）用户数据不丢失

网站数据不能丢失，不论何时何地何种故障，网站的数据都可以快速恢复到最新的状态。

（3）友好的网站体验

服务器的设计一切都是为了所有功能的正常运行，为用户提供一个好的体验，如果你后端数据库优化的再好，前端没用到缓存处理的话，呈现给用户的依然是哪个一卡一顿的页面。

（4）支持集群服务器扩展

好的架构必须支持集群服务器的扩展，利用负载均衡技术和高可技术，可容易地横向扩展服务器，对于未来的用户量的激增也可有效的预防，这是为未来架构的扩展而不影响前三个点所实现的一种需求^[2]。

3 Web 服务端软件性能对比分析

Web 服务端为在线学习用户提供在线信息浏览网页服务。目前，在 Linux 操作系统上提供的 Web 服务软件主要有 Apache、Nginx、Lighttpd 等。

3.1 Apache 服务器

Apache 服务器是一款开源的免费跨平台 Web 服务器软件。它是一个高效、可扩展和安全的 web 服务软件，现在是世界第一大 web 服务器软件，它的最大功能特性是采用了模块化设计，而且该模块化设计分为两大种，一种是 Apache 最常用的静态模块，它是在编译时默认安装的；另外一种的动态扩展模块，可以随时进行安装和卸载，还有具有支持 Unix 线程、支持 ipv6、支持多协议、简化配置、支持 Perl 的正则表达式等优点，这样的灵活性及优点非常适合企业部署适合自己的平台。

3.2 Nginx 服务器

Nginx 是开源 Web 服务器。它的性能极高，稳定性好，功能丰富，配置简单以及需要消耗的资源少。它是专门为高并发场景而设计的，在中国，Nginx 占据了巨大的市场份额，目前国内 BAT 三大巨头的网站都是使用 Nginx 来满足高并发访问的需求。在国外 Nginx 为 WordPress.com、GitHub、Airbnb 等高知名度的网站提供支持。Nginx 采用了最新 I/O 模型，支持并发连接数可达 50000 个。

在线学习平台后端主要采用 Java 和 C/C++ 开发。Apache 与 Tomcat 都是 Apache 开源组织开发的用于处理 HTTP 服务的项目，Apache tomcat 为两者的整合，Apache 服务器负责处理所有静态的页面/图片等信息，Tomcat 处理动态的部分，如果请求是静态网页则由 Apache 处理，并将结果返回；如果是动态请求，Apache 会将解析工作转发给 Tomcat 处理，因此本架构决定采用了 Apache Tomcat 作为 Web 服务器。

4 负载均衡的分析

负载均衡是基于现有的网络架构的，它是一种计算机技术，在互联网服务器上，主要是用于前端 Web 网站和后端数据的负载，在用户发来的请求时，轮询负载均衡后，请求被分发到后端服务器进行处理，然后返回给用户，使学习平台服务端内部网络结构对外部网络用户隐藏，阻止用户直接访问内部网络服务器，服务器更安全。目前实现负载均衡方法主要有两大类：硬件和软件实现。但是为了节约成本，用软件方法来实现。负载均衡可以实现集群服务器的扩展。下面主要介绍三种可以实现负载均衡的软件。

4.1 LVS

LVS（Linux 虚拟服务器）是一种 Linux 虚拟服务，LVS 是一个四层负载均衡，LVS 目前已集成到 Linux 内核模块中，LVS 主要有三种模式八种轮询算法：

① 基于NAT模式。NAT模式即是网络地址转换模式，它是通过LVS负载均衡调度器的外网IP(VIP)映射到内网IP，通过轮询算法将用户的请求分配到内网IP，然后再返回给用户，数据的请求都会通过负载均衡调度器，这样十分用于造成数据的I/O瓶颈。

② 基于TUN模式。TUN模式也叫隧道模式，是将用户的请求和响应数据进行分离。请求的数据是经过负载均衡调度器，响应的数据不需经过负载均衡调度器，而直接通过外网返回给用户，如此所需要消耗大量的外网IP，不推荐使用^[3]。

③ 基于DR模式。DR模式指直接路由模式，它是根据用户发来的报文通过，修改用户的访问的目标地址（原来访问目标是VIP，修改后是轮询后的内网IP）发给内网服务器，然后通过修改源地址（VIP）和目标地址（用户IP），再发回给用户，这样用户觉得自己访问的主机没有变。这是十分典型的四层模式，在建立起TCP连接后，通过修改目标地址和源地址转发数据^[3]。

LVS虽然有其优点，但配置十分繁琐，管理起来也十分困难。

4.2 Nginx反向代理

Nginx反向代理是工作在OSI第七层上，它是通过用户先与Nginx负载均衡先建立起独立的TCP连接，然后Nginx负载均衡与后端服务器连接起独立TCP连接，最后用户传输的数据通过nginx负载均衡轮询后直接转发给后端，nginx只是做代理角色，在数据传输的过程中nginx只做转发功能，这样使得整个网站架构更加智能化，能够把对应的内容转发到对应的服务器，还可以有效防止网络攻击，使网站更加稳定。Nginx反向代理的配置十分简单，而且nginx反向代理十分稳定，对于高并发请求很少出现宕机情况^[4]。

4.3 HAProxy

HAProxy是一个十分强大的开源的负载均衡软件，它是基于四层和七层应用的代理软件。它最主要的特点是性能，支持数以万计的并发连接数，它主要有几种特点：强大的稳定性；支持单机高带宽；支持Web虚拟主机；自带Web监控界面，可以监控负载节点的状态；支持后台节点的监控检查；配置简单；性能强大，比nginx的性能还强大上一筹。

经对比，确定选择了Haproxy作为实现负载均衡的软件。

5 文件服务器的分析

文件服务器是存储文件的重要服务器，它可以通

过网络的形式实现文件共享的。在linux下目前主要有Samba文件服务器，vsftpd服务器，NFS服务器。虽然这三者都可以实现文件共享，但是在工作方式和性能的方面上，Samba和vsftpd远远不及NFS。NFS在速度上更有优势，支持更大的单文件。NFS服务端口支持上百成千台服务器连接共享文件，NFS将端口注册任务委托给RPC来实现，这样NFS更专注于数据共享，这样稳定性也提升了。因此选择NFS文件系统作为后台文件服务器提供存储共享服务。

6 服务器监控的分析

在网站架构搭建完成后，如果直接开始上线使用，在使用过程必会出现各种各样的问题，而监控服务是部署网站架构后必备的服务。监控服务可以实时监控服务器的状态，包括服务器的硬件、服务软件还有业务监控等。有了监控，通过监控软件自动收集信息，并把收集到的信息处理后显示在Web界面，让运维人员就可以有效地处理突发事件，减少突发事件带来的损失^[5]。在linux服务器上的自动化监控主要有三种：Nagios、Cacti以及Zabbix。Nagios算是个古老的监控工具了，它是企业级开源的监控工具，Nagios提供的监控组件包括应用程序、服务、网络协议、操作系统、系统指标和网络基础架构等。数以百计的第三方插件可以监控几乎所有内部和外部应用程序、服务和系统。Nagios的组件十分多，文件配置十分容易出错，对于目前的架构形式，它显得不够高效和使用。Cacti是基于RRDtool的前端主机监控系统，它有可视化的web界面，但界面不够友好，cacti一般通过SNMP获取监控主机的信息存储在mysql上，再通过web界面访问即可，但是对于预处理一些报警事件，不能够及时自动处理。不过对于新手来说这个监控系统容易上手。Zabbix监控系统是一款十分优秀的企业级开源分布式监控系统。它拥有十分丰富的数据采集方式十分友好的web前端界面和灵活的报警处理机制，Zabbix可以监控硬件、系统、网络、应用、业务等。本项目选择Zabbix作为服务器监控软件。

7 在线学习平台集群架构的设计

图1是集群服务器架构逻辑图。集群服务器的工作流程如下：

用户端通过互联网访问到均衡服务器，通过DNS或本地hosts文件解析到架构的前端代理服务器，代理服务器将请求轮询到后面的LNMP web集群服务器上；每台Web服务器都部署了网站内容；当用户更新内容时，会通过Web集群服务器把数据写入分离后的MySQL数据库上；当用户上传图片、附件、头像等，其数据将通过Web集群服务器传到后端的NFS文件服务器上；NFS文件服务器兼职作批量分发和管理服务

器的角色。主要批量分发文件到其他集群服务器上；所有集群服务器的重要文件最终要统一定时备份到备份服务器上保存；评分服务器定期访问数据库服务器，当数据中有学生提交的程序时，将从数据库中取出数据，并实现评分，结果再保存到数据库中，评分服务器与数据库服务器在不同的系统中，确保数据库中数据的安全^[6]。

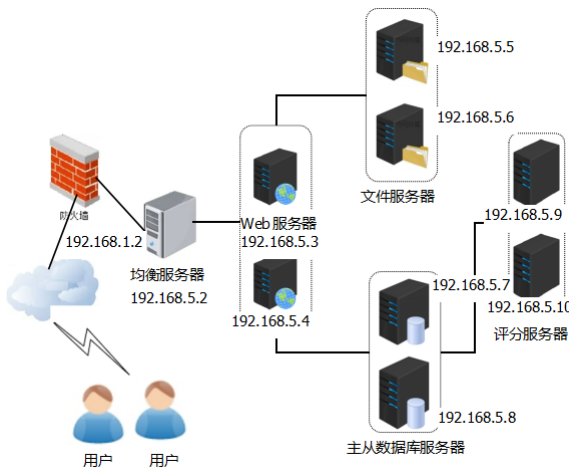


图 1 集群服务器架构逻辑图

8 集群服务器后端关键部分的实现

8.1 主从数据库同步的实现

主机每执行一个操作会在二进制mysql-bin日志文件中记录。从机每隔一段时间就会对主机的日志文件进行探测，如果发生改变，从机就会发送一个IO请求，复制主机中的日志文件进行同步操作，这样从机就可以和主机同步^[7]。具体关键步骤如下：

（1）安装主服务和从服务器上安装mysql数据库，并设置root用户密码。

（2）配置主服务器mysql

```
在/etc/my.cnf配置文件中，打开log-bin记录日志
# vim /etc/my.cnf
log_bin = /data/3306/mysql-bin #开启并自定义日志记录
server_id = 1 #设置唯一识别ID
在主库上进行备份并恢复到从库上。
# mysqldump -uroot -p123456 -x -A -B --events
--master-data=2 > rep.sql
```

（3）配置从服务器

修改my.cnf配置文件，配置如下：

```
# vim /etc/my.cnf
server_id = 102 #mysql的server_id数值必须是唯一的
导入主服务器的数据库
# mysql -uroot -p123456 < rep.sql
登录数据库，开始同步配置
# mysql -uroot -p
mysql> CHANGE MASTER TO
-> MASTER_HOST='192.168.5.7',
```

```
-> MASTER_PORT=3306,
-> MASTER_USER='rep',
-> MASTER_PASSWORD='123456',
-> MASTER_LOG_FILE='mysql-bin.000001',
-> MASTER_LOG_POS=408;
mysql> start slave;
```

8.2 评分服务器连接数据库服务器的实现

（1）通过定义myconnect函数连接指定的Mysql数据库服务器

```
int myconnect(MYSQL *con)
{
    char ip[200];
    getRunParameter("webserverip",ip); //该函数获取服务器的IP并存入ip数组中
    mysql_init(con);
    if(!mysql_real_connect(con,ip,"root","123456","examnew",0,NULL,0))
    {
        printf("连接数据库失败:%s\n",mysql_error(con));
        return 0;
    }
    else
        return 1;
}
```

（2）从Mysql数据库获取数据的关键代码实现

```
ONENODE *getDataFromMysql()
{
    MYSQL con;
    MYSQL_RES *ls;
    MYSQL_ROW hs;
    char cxnr[512];
    int xht;
    ONENODE *onep=NULL;

    if(myconnect(&con))
        sprintf(cxnr,"lock table t_records write,t_answerfile write,t_exam_lib write;");
    if(mysql_real_query(&con,cxnr,(unsigned int)strlen(cxnr)))
    {
        printf("锁定表 t_records,t_answerfile,t_exam_lib 失败!\n");
    }
    /*在这里编写对数据库表操作的sql语句，并存放在 cxnr 数组中 */
    xht=mysql_real_query(&con,cxnr,(unsigned int)strlen(cxnr));
    if(xht)
    {
        printf("查询失败:%s\n",mysql_error(&con));
        return 0;
    }
    ls=mysql_use_result(&con);
    if(hs=mysql_fetch_row(ls))
    {
        /*申请新的结点 */
        onep=(ONENODE *)malloc(sizeof(ONENODE));
        /* 在这里编写语句实现将从数据库中取出来的数据保存到onep结点的成员变量中。*/
        if(strlen(onep->inputdata)==0)
            strcpy(onep->inputdata,"123");
    }
    if(onep!=NULL)
    {

```

```
mysql_free_result(ls);
sprintf(cxn, "update      t_records      set
operatestatus=1, operate time=current_timestamp()
id=%ld;", onep->recordid);
if(mysql_real_query(&con, cxn, (unsigned
int)strlen(cxn)))
    printf("更新状态失败!\n");
}
sprintf(cxn, "unlock tables;");
if(mysql_real_query(&con, cxn, (unsigned int)strlen(cxn)))
    printf("解锁失败!\n");
mysql_close(&con);
}
return onep;
}
```

9 结束语

框架的设计使得用户数据请求的分流,减轻后端单点服务器压力,后端内网服务器:Web服务器集群、NFS服务器集群、数据库服务器集群,通过分析选取了合适学习平台的软件进行搭建,评分服务器,主要是运行用户提交的程序,独立于其它服务器,保证了平台的正常运行及数据的安全;另外,后端服务器对外网是不透明的,外网用户不可以直接访问内网web服务

器,保证内网服务器集群的安全性。架构不仅是有效地解决了目前在线学习平台服务端架构的数据安全及正常运行等问题,也可以为中小型企业提供高可用网站架构部署的参考。

参考文献

- [1] 余洪春.Linux集群和自动化运维[M].北京:机械工业出版社,2016.
- [2] 刘佳.主流WEB服务器搭建技术对比与选择[J].电脑知识与技术,2020,16(15):81-82
- [3] 黑马程序员.Nginx高性能Web服务器实战教程.北京:清华大学出版社,2017
- [4] 李若兰.基于Nginx的Web服务器优化的应用研究[J].科技风,2021(09):119-120
- [5] 郭莹莹.Web服务器的搭建和安全管理探究[J].电子世界,2020(14):109
- [6] 荣露威,陈丽琼.基于Linux平台的Web服务器集群架构设计[J].无线互联科技,2022,19(03):69-70.
- [7] 彭剑,刘艳松,唐闻.MySQL主从服务器数据库同步的实现[J].福建电脑,2020,36(07):118-119.